
Individual Fairness for k -Clustering

Sepideh Mahabadi ^{*1} Ali Vakilian ^{*2}

Abstract

We give a local search based algorithm for k -median and k -means (and more generally for any k -clustering with ℓ_p norm cost function) from the perspective of individual fairness. More precisely, for a point x in a point set P of size n , let $r(x)$ be the minimum radius such that the ball of radius $r(x)$ centered at x has at least n/k points from P . Intuitively, if a set of k random points are chosen from P as centers, every point $x \in P$ expects to have a center within radius $r(x)$. An individually fair clustering provides such a guarantee for every point $x \in P$. In this work, we show how to get a bicriteria approximation for fair k -clustering: The k -median (k -means) cost of our solution is within a constant factor of the cost of an optimal fair k -clustering, and our solution approximately satisfies the fairness condition (also within a constant factor).

1. Introduction

Due to the increasingly use of machine learning in decision-making tasks such as awarding loans, estimating the likelihood of recidivism (Galindo & Tamayo, 2000; Chouldechova, 2017; Dressel & Farid, 2018; Kleinberg et al., 2018), it is crucial to design fair algorithms from the perspective of each individual input entity. In general, the rich area of *algorithmic fairness* over the past few years has had two main aspects, (1) understanding different notions of fairness and formalizing them in the context of learning and optimization tasks (e.g. (Feldman et al., 2015; Kleinberg et al., 2017; Chouldechova & Roth, 2018; Mehrabi et al., 2019; Pessach & Shmueli, 2020)) and (2) designing efficient algorithms with respect to the additional properties caused by the fairness requirement (e.g. (Joseph et al., 2016; Hardt et al., 2016)). Our work in this paper is focused on the latter

direction and in particular the design of fair algorithms for a basic task in unsupervised learning, namely *clustering*. Recently, there has been a large body of work on design of fair algorithms for unsupervised learning tasks and in particular clustering, e.g., (Chierichetti et al., 2017; Bera et al., 2019; Abraham et al., 2019; Har-Peled & Mahabadi, 2019; Elzayn et al., 2019; Baharlouei et al., 2019).

Clustering is a fundamental task with huge number of applications such as feature engineering, recommendation systems and urban planning. Due to its importance, the clustering problem has been studied extensively from the fairness point view over the past few years (Chierichetti et al., 2017; Rösner & Schmidt, 2018; Kleindessner et al., 2019a;b; Backurs et al., 2019; Chen et al., 2019; Schmidt et al., 2019; Bercea et al., 2019; Bera et al., 2019; Huang et al., 2019; Ahmadian et al., 2019; Davidson & Ravi, 2019; Abraham et al., 2019). However, most previous results on this topic consider the clustering problem with respect to the notion of *group fairness*. As introduced by (Chierichetti et al., 2017), in the clustering problem with respect to the *group fairness* requirement, the high-level goal is to come up with a minimum cost clustering of a given set of points with an extra constraint that requires all clusters to be balanced with respect to a set of specified protected attributes such as gender, race or religious.

In this paper, following the work of (Jung et al., 2019), we study the clustering problem from the *individual fairness* point of view: The goal is to design a clustering of the input point set so that all points are treated (approximately) equally. As an example, this is important when the clustering is used in certain infrastructural decisions such as where to open new facilities to serve residents in different neighborhoods. Formally, given a point set P of size n , the *fair radius* is defined for each point $p \in P$ as the minimum radius such that the ball $B(p, r(p))$ contains at least $\lceil n/k \rceil$ points from P . Intuitively, this is the radius which the point p expects to have a center within, if the centers were to be chosen uniformly at random. Therefore, it is natural to ask for a clustering solution to approximately respects this expectation and provides a clustering that has a center within distance $O(r(p))$ for every point $p \in P$, thus providing an individually fair clustering solution.

As mentioned above, this notion of fair clustering was in-

^{*}Equal contribution ¹TTIC, Chicago, Illinois, USA ²University of Wisconsin, Madison, Wisconsin, USA. Correspondence to: Sepideh Mahabadi <mahabadi@ttic.edu>, Ali Vakilian <vakilian@mit.edu>.

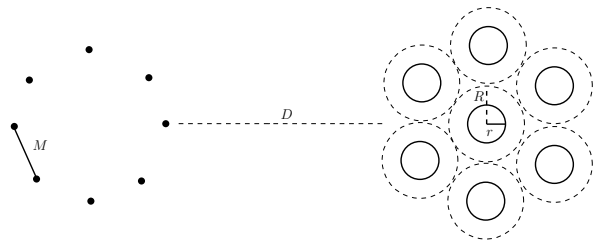


Figure 1.1. This example shows that for any values of α and k , an optimal solution of k -median (or any other ℓ_p norm cost functions such as k -means or k -center) can be arbitrarily unfair. This is described in the proof of Observation 1.1 in Appendix A.1. In this example, M denote the pairwise distance of points in left part and D denotes the minimum distance of a point in the right part to a point in the left part. Moreover, the parameters r , R , M and D are picked so that $R \gg r$ and $M = D \gg 2n(R + r)$.

troduced in (Jung et al., 2019) where the authors showed that it is possible to get a 2-approximate fair clustering, meaning that there exists a feasible solution of a set of k centers where every point p in the input has a center within distance $2r(p)$. This algorithmic result is based on the previous works of (Chan et al., 2006; Charikar et al., 2010) on metric embedding. Among other results, they showed that this factor of two loss in the fairness is tight: there are metric spaces and configurations of points where for $\alpha < 2$ it is impossible to find k -centers such that for every point p , the distance of p to its center is at most $\alpha r(p)$. Moreover, they showed empirically that the standard k -median (k -means) algorithm does not provide a good fairness guarantee and further demonstrated the *price of fairness* on real-world data sets.

1.1. Our Contribution

The above empirical result is confirmed by the following observation which shows that the solution returned by existing standard approaches for k -clustering that are oblivious to this fairness requirement, may be arbitrarily far from being fair.

Observation 1.1. *The fairness ratio of an optimal k -clustering can be arbitrarily large. (Proof is in Appendix A.1)*

This in particular shows the need to design “efficient” algorithms with this notion of fairness.

In this work, we show how to find a fair clustering that also minimizes the k -median or k -means cost (and more generally, any ℓ_p norm cost functions with $p \geq 1$). More specifically, we show that a variant of the local search algorithm provides the following $(O(1), O(1))$ -bicriteria approximation for fair k -median and k -means.

Theorem 1.2. *Given any desired fairness parameter $\alpha \geq 1$,*

there exists a polynomial time constant factor bicriteria approximation algorithm: The algorithm can find a set of centers $\mathcal{C} \subseteq P$ of size k , such that for each point $p \in P$, we have $d(p, \mathcal{C}) \leq O(\alpha \cdot r(p))$, and further, $\text{cost}(\mathcal{C}) \leq O(\text{cost}(\text{OPT}_\alpha))$. Here $\text{cost}(\text{OPT}_\alpha)$ denotes the minimum clustering cost of P with k centers such that for every point p , there exists a center within distance $\alpha \cdot r(p)$.

We remark that while in this paper we only mention k -means and k -median clustering, more generally in Appendix B, we show that our analysis provides $(O(1), O(p))$ -approximation for any cost function of the form $(\sum_{x \in P} d(x, S)^p)^{1/p}$; this in particular implies $(O(1), O(1))$ -approximation for α -fair k -median (when $p = 1$), $(O(1), O(1))$ -approximation for α -fair k -means (when $p = 2$) and $(O(1), O(\log n))$ -approximation for α -fair k -center (when $p = \log n$).

Also, we again note that our result is in contrast to the previous result which only provided one (approximately) feasible fair clustering *without minimizing the clustering cost*. Our algorithm is based on the local search algorithm and is easy to implement. More precisely, we start with a feasible solution which combines the output of the described algorithm of (Chan et al., 2006; Charikar et al., 2010; Jung et al., 2019) with the standard greedy algorithm of k -center (described in Section 3.2 in more details). Then in successive iterations, the algorithm improves the k -median (k -means) cost while respecting the fairness condition. Moreover, in order to show the theoretical guarantee, the local search algorithm should take swaps of size 4, meaning that it considers swapping of at most 4 centers in its current solution with the points outside of the solution.

Although local search is a widely used approach for k -median (k -means) clustering, our analysis is more involved and involves new structures that are crucial for handling the fairness constraints. We remark that for the sake of simplicity and readability of the paper, we have not optimized the constants in the approximation factors of the cost and the fairness guarantees.

Experiments. Further, we run experiments on three datasets (Diabetes, Bank, Census) that have been previously used in the context of fair clustering (e.g., see (2017; 2019; 2019; 2019; 2019)). Our experiments show that in compare to the algorithm of (Jung et al., 2019), the k -median cost improves on average by a factor of 1.86, but it loses on fairness by a factor of 1.26 on average.

2. Preliminaries

Throughout the paper, we use P to denote the set of points that we wish to cluster, and the parameter k to denote the number of centers we allow for clustering. For each

$x \in P$, we use $B(x, r) = \{y \in P : d(x, y) \leq r\}$ to denote the set of points that are contained in the ball of radius r around x . Also, in this paper we consider two main variants of clustering costs, k -median and k -means. In k -median, the goal is to select k centers in P such that the total sum of distances of points to their centers is minimized, $\min_{S \subseteq P: |S| \leq k} \sum_{p \in P} d(p, S)$, where for a set of points S and a point p , the distance $d(p, S)$ is defined to be $\min_{s \in S} d(p, s)$. In k -means, the goal is to select k centers in P such that the sum of the square distances of points to their centers is minimized, $\min_{S \subseteq P: |S| \leq k} \sum_{p \in P} d(p, S)^2$. We also use $d(x, y)$ to denote $d(x, y)^2$ when working under the k -means cost.

More generally, we analyze our algorithm for the fair variant of clustering with any ℓ_p norm cost function, $\min_{S \subseteq P: |S| \leq k} (\sum_{p \in P} d(p, S)^p)^{1/p}$, where $p \geq 1$. Besides including k -means and k -median as its special cases, the general ℓ_p norm cost function implies an approximation guarantee for another common clustering cost function, namely k -center. In standard k -center, the goal is to minimize the maximum distance of points to their centers: $\min_{S \subseteq P: |S| \leq k} \max_{p \in P} d(p, S)$.

Next, we formally define a fair radius for each point in the point set.

Definition 2.1 (fair radius). *Given a set of n points P in a metric space (X, d) and a parameter $\ell \in [n]$, for each $x \in P$ define $r_\ell(x)$ to be the radius of the minimum ball centered at x that contains (n/ℓ) points of P ; $r_\ell(x) = \min\{r : |B(x, r)| \geq n/\ell\}$.*

Definition 2.2 (α -fair clustering (Jung et al., 2019)).

Given a set of n points P in a metric space (X, d) , a k -clustering using a set of centers S is (α, ℓ) -fair if for any $x \in P$, $d(x, S) \leq \alpha \cdot r_\ell(x)$ where $d(x, S)$ denotes the distance of x to its closest neighbor in S . In the case $\ell = k$, we succinctly denote it as α -fair k -clustering.

Definition 2.3 (bicriteria approximation). *Given a set of points P in a metric space (X, d) , an algorithm is a (β, γ) -approximation for α -fair k -clustering of a given cost function cost^1 if the solution SOL returned by the algorithm satisfies the following properties:*

1. **Cost guarantee:** $\text{cost}(\text{SOL}) \leq \beta \cdot \text{cost}(\text{OPT}_\alpha)$ where OPT_α denotes an optimal solution of α -fair clustering with respect to the given cost function cost , and
2. **Fairness guarantee:** SOL is a $(\gamma \cdot \alpha)$ -fair k -clustering.

Next, we define *critical balls* which are crucial in our analysis of the local search algorithm. The notion is used to show that our solution satisfies the fairness guarantee.

¹ k -median, k -means, k -center or more generally any ℓ_p norm cost function.

Definition 2.4 (critical balls). *Given a collection of n points P in a metric space (X, d) , a set of balls $B_1 = B(c_1^*, \alpha r_k(c_1^*)), \dots, B_\ell = B(c_\ell^*, \alpha r_k(c_\ell^*))$ (where $\ell \leq k$) are called critical if they satisfy the following properties:*

C-1 For each $x \in P$, $d(x, \{c_1^*, \dots, c_\ell^*\}) \leq 6\alpha r_k(x)$,

C-2 For any pair of centers c_i^* and c_j^* , $d(c_i, c_j) > 6\alpha \max\{r_k(c_i), r_k(c_j)\}$

In Lemma 4.1, we show that there exists a polynomial time algorithm for finding a set of critical balls of P . We say that a set of centers S is *feasible* with respect to a set of given critical balls \mathcal{B} if for each $B \in \mathcal{B}$, $|B \cap S| \geq 1$; each critical ball contains a center from S .

Claim 2.5. *Let o be a point in a critical ball $B \in \mathcal{B}$. The nearest neighbor of o in a given set S of feasible centers with respect to the critical balls cannot belong to a ball other than B . Moreover, the nearest neighbors of two points $o_1 \in B_1$ and $o_2 \in B_2$ where $B_1 \neq B_2$ cannot be the same.*

Proof: Consider the critical ball $B_1 = B(c_1, \alpha r_k(c_1))$ that contains o_1 . Since S is a feasible center set with respect to \mathcal{B} , $d(o_1, \text{NN}_S(o_1)) \leq 2\alpha r_k(c_1)$. However the distance of o_1 to any other critical ball $B_2 = B(c_2, \alpha r_k(c_2))$ is at least $d(c_1, c_2) - \alpha r_k(c_1) - \alpha r_k(c_2) > 4\alpha r_k(c_1)$ using Definition 2.4. Therefore, the nearest neighbor of o_1 cannot be in any ball other than B_1 .

Now if two points $o_1 \in B_1$ and $o_2 \in B_2$ are in different balls and have the same nearest neighbor $s \in S$, it means that their distance is at most $d(o_1, s) + d(o_2, s) \leq 2\alpha r_k(c_1) + 2\alpha r_k(c_2) \leq 4\alpha \max\{r_k(c_1), r_k(c_2)\}$. However by the previous argument their distance is larger than $4\alpha r_k(c_1)$ which is a contradiction. \square

Lastly, for a point x , let $\text{NN}_Y(x)$ denote the nearest neighbor of x in the set of points Y . When $Y = P$, we drop the subscript and refer to NN_P as NN .

3. High Level Description of Our Algorithm

In this section, we provide a high-level description of our local search algorithm for α -fair k -median. We note that the algorithm for α -fair k -clustering with other cost functions (e.g., k -means or more generally any ℓ_p norm cost) is almost identical but in Section 3.2, where we compute the clustering cost in the local search algorithm, instead of working with pairwise distances we consider the corresponding function of distance (e.g., distance squared for k -means).

3.1. Handling Fairness Constraints via Critical Balls

We use a slightly modified variant of the greedy approach of (Chan et al., 2006; Charikar et al., 2010) to find $\ell \leq k$

disjoint *critical balls* (see Lemma 4.1). Then, we ignore the fairness constraint and instead run the local search algorithm with an extra requirement: Find a set of k centers that are feasible with respect to the critical balls. By Lemma 4.2 in Section 4, these centers result an $O(\alpha)$ -fair k -clustering.

Algorithm 1 Finds a set of critical balls of size at most k

```

1: Input: set of points  $P$  of size  $n$ , fairness parameter  $\alpha$ 
2:  $Z \leftarrow P, C^* \leftarrow \emptyset$ 
3: repeat
4:    $c \leftarrow \operatorname{argmin}_{x \in Z} r_k(x)$ 
5:    $C^* \leftarrow C^* \cup \{c\}$ 
6:    $Z \leftarrow \{x \in Z : d(x, c) > 6\alpha \cdot r_k(x)\}$ 
7: until  $Z \neq \emptyset$ 
8: return  $\{B(c, \alpha r_k(c)) : c \in C^*\}$ 
    
```

3.2. Initialization and Local Search Update

Next, we initiate the algorithm with a feasible set of centers S_0 with respect to the constructed set of critical balls \mathcal{B} by Algorithm 1. Note that the choice of initial feasible set of centers plays an important role in bounding the number of iterations required by our local search algorithm and consequently the total runtime of our algorithms. In Theorem 3.1, we show that a modified variant of the standard greedy algorithm of k -center can be used to find a good initial set of centers S_0 . See part **I** in Algorithm 2.

Then, we go through iterations and in each iteration j , we check whether there exists a swap of size at most t (i.e., replacing $t' \leq t$ centers in the current center set S_j with a set of t' centers outside of S_j) that results in a feasible set of centers S' with respect to \mathcal{B} whose clustering cost improves upon the clustering cost with S_j as centers by a factor more than $1/(1 - \varepsilon)$, i.e., $\operatorname{cost}(S') \leq (1 - \varepsilon) \cdot \operatorname{cost}(S_j)$. If there exists such a set S' , then we set $S_{j+1} = S'$ and proceed to the next iteration; otherwise, we stop the process and output the current set of centers, S_j , as our solution. We refer to such S_j as a set of (t, ε) -stable centers; there is no feasible swaps of size at most t with respect to the critical balls that improve the clustering cost “significantly”. See part **II** in Algorithm 2.

Theorem 3.1. *The local search algorithm stops after $O(\frac{\log n}{\varepsilon})$ iterations. Moreover, the runtime of the algorithm is $\tilde{O}((kn)^{t+1} \cdot \varepsilon^{-1})$.*

Proof: Let $C^* = \{c_1^*, \dots, c_\ell^*\}$ denote the set of the balls returned by Algorithm 1. To construct S_0 , we run the following modified greedy algorithm of k -center (see part **I** in Algorithm 2).

Initialize S_0 to C^* . Then, go through $k - \ell$ rounds and in each round i , add to S_0 the point $x_i \in P \setminus S_0$ to S_0 who is the furthest from S_0 ; $\forall x \in P \setminus S_0, d(x_i, S_0) \geq d(x, S_0)$.

Algorithm 2 local search algorithm w.r.t. critical balls

```

1: Input: set of points  $P$  of size  $n$ , set of critical balls  $\mathcal{B}$ 
   along with their centers  $C^*$ , upper bound  $t$  on swap size
2: I. Constructing Initial Center Set  $S'$ 
3:  $S' \leftarrow C^*$ 
4: for  $i = 1$  to  $k - \ell$  do
5:    $z \leftarrow \operatorname{argmax}_{x \in P \setminus S'} d(x, S')$ 
6:    $S' \leftarrow S' \cup \{z\}$ 
7: end for
8: II. Local Search Update
9: repeat
10:   $S \leftarrow S'$ 
11:  for  $i = 1$  to  $t$  do
12:    for  $T_1 \subseteq S$  and  $T_2 \subseteq P \setminus S$  of size  $i$  do
13:      if  $(S \cup T_2) \setminus T_1$  is feasible w.r.t  $\mathcal{B}$  then
14:         $S' \leftarrow (S \cup T_2) \setminus T_1$ 
15:        if  $\operatorname{cost}(S') \leq (1 - \varepsilon) \cdot \operatorname{cost}(S)$  then
16:          break to line 21
17:        end if
18:      end if
19:    end for
20:  end for
21: until  $\operatorname{cost}(S') \leq (1 - \varepsilon) \cdot \operatorname{cost}(S)$ 
22: return  $S$ 
    
```

When the greedy algorithm terminates, let μ denote the maximum distance of a point in $P \setminus S_0$ to S_0 ; $\mu := \max_{x \in P \setminus S_0} d(x, S_0)$. Note S_0 is a feasible set of centers with respect to the critical balls \mathcal{B} and the clustering cost with S_0 as centers is at most $n \cdot \mu$. Next, we show that the cost of k -clustering with any feasible set of centers with respect to the critical balls (in particular, the set of optimal centers O) is at least $\Omega(\mu)$. Consider the $k + 1$ points in $T := S_0 \cup \{x\}$. In any k -clustering at least two points p_1, p_2 in T belong to a same cluster. There are two cases for p_1 and p_2 :

- *At least one of p_1, p_2 belongs to $T \setminus C^*$.* Let assume that p_2 denote the one that added to T later than p_1 (the last point added to T is x). Note that by the description of the greedy process, for each x_i , $d(x_i, C^* \cup \{x_1, \dots, x_{i-1}\}) \geq d(x, S_0) = \mu$. This implies that $d(p_1, p_2) \geq \mu$. Then, since $d(\cdot)$ satisfies the triangle inequality², the distance of at least one of p_1 and p_2 to the center of their cluster is $\Omega(\mu)$. Hence, $\operatorname{OPT} = \Omega(\mu)$.
- *Both p_1, p_2 belong to C^* .* Since all points in C^* belong to different critical balls, by Claim 2.5, in any feasible

²Note that the squared distance (which is used in k -means) does not satisfy the triangle inequality but instead satisfy an approximate version of triangle inequality which is sufficient for our purpose: $d(x, y)^2 \leq 2d(x, z)^2 + 2d(z, y)^2$.

set of centers with respect to \mathcal{B} , C^* belong to different clusters. Hence, this case cannot happen.

We showed that $\text{cost}(S_0) = O(n \cdot \text{cost}(O)) = O(n \cdot \text{OPT})$. Since, in each round of the local search algorithm the cost decrease by a factor of $(1 - \varepsilon)$, the total number of rounds before obtaining a t -stable set of centers is $O(\frac{\log n}{\varepsilon})$. Moreover, the time to decide whether there exist a swap of size at most t that improves the clustering cost by at least a factor of $(1 - \varepsilon)$ is $O(k^t \cdot n^t \cdot n \cdot k)$ where $O(k^t \cdot n^t)$ bounds the number of swaps of size at most t and kn denotes the required amount of time to recompute the clustering cost for each potential swap. In total, the algorithm runs in time $\tilde{O}((kn)^{t+1} \cdot \varepsilon^{-1})$. \square

In Sections 4 and 5 we prove that the algorithm provides a bicriteria approximation. All missing proofs of the paper are presented in the appendix (A.2-A.5).

4. Handling Fairness Constraints

To satisfy the fairness requirement, as a first step, by slightly modifying the greedy algorithm of (Chan et al., 2006; Charikar et al., 2010), in Lemma 4.1 we show that given a set of points P we can find a set of critical balls in polynomial time. Then, in Lemma 4.2 we show that a feasible set of centers with respect to the critical balls is an approximate α -fair solution.

Lemma 4.1. *Given a set of n points P in a metric space (X, d) , there is an algorithm that runs in $O(n^2)$ and finds a set of critical balls of size at most k .*

Next, we show that in order to provide the fairness guarantee (approximately), it suffices to only satisfy the guarantee for the subset of points generated by Algorithm 1. In particular, this reduces the problem of α -fair k -clustering to an instance of k -clustering with *partition* constraints: given a collection of points P and a collection of ℓ critical balls $\mathcal{B} = \{B_1, \dots, B_\ell\}$ where $\ell \leq k$, find a minimum cost clustering which is feasible with respect to \mathcal{B} .

Note that by the definition of α -fairness, it is straightforward to verify that an α -fair k -clustering of P is a feasible k -clustering with respect to the critical balls \mathcal{B} . In the following lemma we prove that any feasible k -clustering with respect to the critical balls is $O(\alpha)$ -fair.

Lemma 4.2. *Given a set of $\ell \leq k$ of critical balls \mathcal{B} centered at C^* of the input point set P , let $S = \{s_1, \dots, s_k\}$ be a feasible k -clustering solution with respect to \mathcal{B} . Then, the corresponding clustering using S as centers is an $O(\alpha)$ -fair k -clustering of P .*

5. Analysis of Local Search Algorithm for α -Fair k -Median

In this section, we analyze our proposed *local search* algorithm for the α -fair k -median clustering. In Section 5.2, we adopt the analysis of the local search algorithm for the “vanilla” k -median (Arya et al., 2004) to the fair k -median problem via the mapping and covering introduced in Section 5.1 and show that the local search approach achieves an $(O(1), O(1))$ -approximation for α -fair k -clustering. Similarly, in Appendix B, we follow the local search analysis of (Kanungo et al., 2004; Gupta & Tangwongsan, 2008) via our new mapping and covering constructions to prove an $(O(1), O(1))$ -approximation for α -fair k -means and more generally any ℓ_p norm cost function.

5.1. Bounded Mapping and Covering

The analysis of the local search of “vanilla” k -median (and similarly k -means) relies on the existence of a mapping between any set of (t, ε) -stable centers and the set of optimal centers.

We use O and S to respectively denote an optimal set of centers for α -fair k -median of P and a set of (t, ε) -stable centers of P with respect to the critical balls \mathcal{B} . Note that since not all sets of k points in P are feasible centers with respect to \mathcal{B} , we have to deal with extra constraints when we design a mapping $\pi : O \rightarrow S$.

Next we list the desired properties for the mapping π and introduce a covering \mathcal{Q} of the edges in π with certain properties that help us to bound the approximation guarantee of our algorithm. For simplicity, we assume that the sets S and O are disjoint. We remark that if $S \cap O$ is non empty, then we can simply ignore the centers in $S \cap O$ along with their clusters in the optimal solution at no cost.

Definition 5.1 (Δ -bounded Mapping). *Given a pair of sets of points S, O in a metric space (X, d) , we say a mapping $\pi : O \rightarrow S$ is Δ -bounded if it satisfies the following properties:*

- D-1 *Each center in O is mapped to exactly one center in S .*
- D-2 *For all $s \in S$, at most Δ centers in O are mapped to s .*

Definition 5.2 ((t, γ) -bounded Covering). *Let $\mathcal{B} = \{B_1, \dots, B_\ell\}$ be a set of critical balls for P . We define $\mathcal{Q} = \{Q_1, \dots, Q_m\}$ as a covering of all edges in $\pi : O \rightarrow S$, where each Q_i is a subset of the edges $(o, \pi(o))$ such that their union covers all edges of the mapping. We refer to each Q_i as a partition. For each partition $Q \in \mathcal{Q}$, let $O(Q)$ denote the set of endpoints in Q that belong to O , and let $S(Q)$ denote the set of endpoints in Q that belong to S . We say that the covering \mathcal{Q} is (t, γ) -bounded if it satisfies the following properties:*

E-1 Each edge $(o, \pi(o))$ appears in at least one and at most γ partitions in \mathcal{Q} .

E-2 Each partition Q is a set of at most t disjoint edges. In other words, (i) for each pair of $o_1 \neq o_2 \in O(Q)$, $\pi(o_1) \neq \pi(o_2)$ and (ii) $|O(Q)| = |S(Q)| \leq t$.

E-3 For each partition $Q \in \mathcal{Q}$ and each critical ball $B_j \in \mathcal{B}$, $|((B_j \cap S) \setminus S(Q)) \cup (B_j \cap O(Q))| \geq 1$. In words, by performing the swaps corresponding to the edges in Q , each ball B_j will have at least one center in $(S \setminus S(Q)) \cup O(Q)$.

E-4 Given a partition Q , for all pair $o \in O(Q)$ and $o' \in O \setminus O(Q)$, $\pi(o) \neq \text{NN}(o')$.

5.2. Analysis of Local Search

In Sections 5.3 and 5.4, we prove the main result of this section which says that given a pair of sets of centers O and S , we can find an $O(1)$ -bounded mapping π together with a $(O(1), O(1))$ -bounded covering \mathcal{Q} of the edges in π . In the following we show that if there exist such bounded mapping and covering, then the k -median cost of P using a (t, ε) -stable set of centers S is within a constant factor of an optimal α -fair k -median of P (i.e., clustering using O).

Lemma 5.3. Consider a set of points P in a metric space (X, d) and let S be a set of $(t, \varepsilon = (1/2tk))$ -stable centers. Suppose that there exists a pair of Δ -bounded mapping $\pi : O \rightarrow S$ and (t, γ) -bounded covering \mathcal{Q} of the edges in π . Then, $\text{cost}(S) \leq 2\gamma \cdot (2\Delta + 1) \cdot \text{OPT}$ where OPT denotes the cost of an optimal α -fair k -median of P (i.e., the k -median cost using O).

5.3. Construction of Mapping π

Let us start by a few notations and claims that we will use to construct our mapping.

Notations. We use \mathcal{NN} to denote the subset of points in S that are the nearest neighbors of a point in O ; $\mathcal{NN} = \{s \in S : \exists o \in O \text{ s.t. } \text{NN}_S(o) = s\}$. Moreover, for any value of $i \geq 0$, \mathcal{NN}_i denotes the subset of S that are the nearest neighbors of exactly i points in O ; $\mathcal{NN}_i = \{s \in S : |\{o \in O : \text{NN}_S(o) = s\}| = i\}$. We also write $\mathcal{NN}_{\geq i}$ to denote the set $\{s \in S : |\{o \in O : \text{NN}_S(o) = s\}| \geq i\}$.

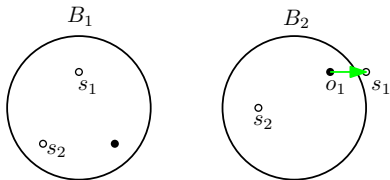


Figure 5.1. B_1 and B_2 are safe balls.

Definition 5.4 (Safe Ball/Point/Edge). A ball $B \in \mathcal{B}$ is safe if

1. either it has at least two points from S ,
2. or it contains a point $o \in O$ that has a unique nearest neighbor (i.e., $\text{NN}_S(o) \in \mathcal{NN}_1$).

A point $s \in S$ is safe if $s \notin \mathcal{NN}_{\geq 2}$ and

1. either s is not contained in any ball of \mathcal{B} ,
2. or the ball containing s is safe.

Moreover, for a pair of vertices $o \in O$ and $s \in S$ we say that the edge (o, s) is safe if

1. either s is a safe point,
2. or both s and o belong to the same ball in \mathcal{B} .

Lastly, when a ball or point or edge is not safe we denote it as unsafe.

Definition 5.5 (s_{NN} and s_{in}). For each ball $B \in \mathcal{B}$, define $s_{\text{NN}}(B)$ to be the point in S that is the nearest neighbor of an arbitrary point $o \in B \cap O$. Similarly, for each ball $B \in \mathcal{B}$, let $s_{\text{in}}(B)$ be an arbitrary point $s \in B \cap S$.

Observation 5.6. For any pair of balls $B_1, B_2 \in \mathcal{B}$, $s_{\text{NN}}(B_1) \neq s_{\text{NN}}(B_2)$.

The observation above follows from Claim 2.5.

Observation 5.7. The set of unsafe points that are in \mathcal{NN}_0 is a subset of $\bigcup_{B \in \mathcal{B}} s_{\text{in}}(B)$.

Observation 5.8. The number of points in S that are not the nearest neighbor of any point in O is large enough, i.e.,

$$|\mathcal{NN}_0| = \sum_{s \in S} \max\{0, |\text{NN}^{-1}(s)| - 1\}. \quad (5.1)$$

The following lemma shows that there exist enough safe points in \mathcal{NN}_0 .

Lemma 5.9. Let \mathcal{F} denote the set of safe points in \mathcal{NN}_0 . Then,

$$|\mathcal{F}| \geq \sum_{s \in S} \max\{0, |\text{NN}^{-1}(s)| - 2\}. \quad (5.2)$$

Next we will define a mapping $\pi : O \rightarrow S$ from which we can later derive a covering \mathcal{Q} .

Lemma 5.10 (3-bounded Mapping). There exists a mapping $\pi : O \rightarrow S$ with the following properties:

1. If $\text{NN}_S(o) \in \mathcal{NN}_1$, then $\pi(o) = \text{NN}_S(o)$.

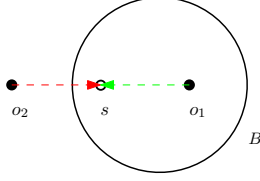


Figure 5.2. B is an unsafe ball and $s = \pi(o_1) = \pi(o_2)$. Moreover, s is not the nearest neighbor of any points other than o_1 and o_2 .

2. If $\text{NN}_S(o) \in \mathcal{NN}_{\geq 3}$, then the edge $(o, \pi(o))$ is safe and $\pi(o) \in \mathcal{NN}_0$.
3. If s is the nearest neighbor of exactly two distinct points o_1, o_2 in O , then $\pi(o_1) = \pi(o_2) \in \mathcal{NN}_0$. Moreover, one of the following holds
 - (a) each of the edges $(o_i, \pi(o_i))$ is safe.
 - (b) each of the edges $(o_i, \pi(o_i))$ is unsafe and goes to a ball that has a safe outgoing edge.
 - (c) there exists an (unsafe) ball $B \in \mathcal{B}$ such that $(o_1, \pi(o_1))$ belongs to B and o_2 does not belong to any ball (see Figure 5.2).
4. For each s , $|\{o \in O : \pi(o) = s\}| \leq 3$.

Proof: We design π as follows.

Step 1: $\text{NN}_S(o) \in \mathcal{NN}_1$. In this case we set $\pi(o) = \text{NN}_S(o)$. This guarantees the first property in the lemma.

Step 2: $\text{NN}_S(o) \in \mathcal{NN}_{\geq 3}$. By Lemma 5.9 there exist enough safe points in \mathcal{NN}_0 so that for each point $s \in \mathcal{NN}_{\geq 3}$, we can allocate a set of safe points $F(s)$, such that i) $|F(s)| = |\text{NN}^{-1}(s)| - 2$, and ii) $F(s) \cap F(s') = \emptyset$ for any $s' \neq s$. Next we define the mapping π for the set of points in $|\text{NN}^{-1}(s)|$ and map them to the points in $F(s)$ such that the in-degree of each safe point in $F(s)$ is at most $|\text{NN}^{-1}(s)| / (|\text{NN}^{-1}(s)| - 2) \leq 3$. Note that for each $o' \in \text{NN}^{-1}(s)$, $(o', \pi(o'))$ is a safe edge. This guarantees the second property in the lemma.

Claim 5.11. Let \mathcal{F}_0 denote the set of points in \mathcal{NN}_0 for which some points in O are mapped to by the end of step 2. Then, $|\mathcal{NN}_0 \setminus \mathcal{F}_0| \geq |\mathcal{NN}_{\geq 2}|$.

Proof: The step 2 of the construction of the mapping consumes exactly $\sum_{s \in \mathcal{S}} \max(0, |\text{NN}^{-1}(s)| - 2)$ safe points in \mathcal{NN}_0 ; $|\mathcal{F}_0| = \sum_{s \in \mathcal{S}} \max(0, |\text{NN}^{-1}(s)| - 2)$. On the other hand, by Observation 5.8, $|\mathcal{NN}_0| = \sum_{s \in \mathcal{S}} \max(0, |\text{NN}^{-1}(s)| - 1)$. Hence, $|\mathcal{NN}_0 \setminus \mathcal{F}_0| \geq |\mathcal{NN}_{\geq 2}|$. \square

Step 3.a: $\text{NN}_S(o) \in \mathcal{NN}_2$. First we assign these points to unused safe points (if there exists any) in a way so that if $\text{NN}(o_1) = \text{NN}(o_2)$ then $\pi(o_1) = \pi(o_2)$. In this case

both $(o_1, \pi(o_1))$ and $(o_2, \pi(o_2))$ are safe and the degree of $\pi(o_1) = \pi(o_2)$ is exactly 2 which corresponds to property (3a) in the lemma.

Note that by Claim 5.11, at the end of step 2, the number of free points in \mathcal{NN}_0 is at least $|\mathcal{NN}_{\geq 2}|$ and we have enough free points in \mathcal{NN}_0 for mapping the subset of points whose nearest neighbors belong to \mathcal{NN}_2 .

Claim 5.12. All outgoing edges of the mapping π constructed so far (i.e., in step 1, 2 and 3.a) that leave a ball in \mathcal{B} are safe.

Proof: All edges of π constructed in steps 2 and 3.a are safe as stated earlier. So consider an outgoing edge (o, s) in π where $s = \text{NN}_S(o)$ belongs to \mathcal{NN}_1 . By Claim 2.5, s cannot be in any ball and thus it is a safe point and hence (o, s) is safe. \square

Next, let $\overline{\mathcal{NN}}_2$ denote the set of points $s \in \mathcal{NN}_2$ such that the mapping π is not yet defined for o_1, o_2 where $\text{NN}_S(o_1) = \text{NN}_S(o_2) = s$. Moreover, let $\overline{\mathcal{NN}}_0$ denote the subset of \mathcal{NN}_0 that are not yet used in the mapping π so far. Since for each pair o_1, o_2 where $\text{NN}(o_1) = \text{NN}(o_2) \in \mathcal{NN}_2$, $\pi(o_1) = \pi(o_2)$, it is straightforward to verify that the invariant $|\overline{\mathcal{NN}}_0| \geq |\overline{\mathcal{NN}}_2|$ still holds.

Step 3.b: $\text{NN}_S(o) \in \overline{\mathcal{NN}}_2$.

1. For a pair of points o_1 and o_2 whose (identical) nearest neighbor belongs to $\overline{\mathcal{NN}}_2$, if there exists an unsafe ball in \mathcal{B} that contains a free point $s \in \overline{\mathcal{NN}}_0$ (i.e., is not assigned to any point in O by the mapping π so far) and \mathcal{B} contains at least one of o_1 and o_2 then set $\pi(o_1) = \pi(o_2) = s$. (Note that using Claim 2.5, o_1 and o_2 cannot belong to different balls as they share the same nearest neighbor.)
2. Once there is no such pair of points o_1 and o_2 as in the previous case anymore, we consider an arbitrary one-to-one assignment ϕ from the free points in $\overline{\mathcal{NN}}_2$ to $\overline{\mathcal{NN}}_0$. Finally, for a pair of points o_1 and o_2 where $\text{NN}_S(o_1) = \text{NN}_S(o_2) = s \in \overline{\mathcal{NN}}_2$ we set $\pi(o_1) = \pi(o_2) = \phi(s)$.

Next we show that if $(o_1, \pi(o_1) = s_1)$ is an unsafe edge (which is true for all constructed edges in step 3.b), then either property (3b) or (3c) holds. Suppose for contradiction that (o_1, s_1) goes to a ball B that does not have any safe outgoing edge and $o_2 \notin B$. Note that by the feasibility of centers set O , the ball B contains a point in $o \in O$ where $o \notin \{o_1, o_2\}$. Since (o_1, s_1) is an unsafe edge, $B \cap S = \{s_1\}$ which implies that $\pi(o)$ does not belong to B . Moreover, by our construction (step 3.b-1 above) $\text{NN}_S(o) \notin \overline{\mathcal{NN}}_2$; otherwise, o had to be mapped to s_1 . Hence, by Claim 5.12, the outgoing edge $(o, \pi(o))$ is safe which is a contradiction. \square

5.4. Construction of the Covering \mathcal{Q}

Lemma 5.13 ((4, 6)-bounded Covering). *Given a mapping π that satisfies conditions of Lemma 5.10, we can find a (4, 6)-bounded covering \mathcal{Q} .*

Corollary 5.14 (fair k -median). *The local search algorithm with swaps of size at most 4 returns a $(84, 7)$ -bicriteria approximate solution of α -fair k -median of a point set P of size n in time $\tilde{O}(k^5 n^5)$.*

Proof: By Lemma 4.2, the result of our local search algorithm returns a (7α) -fair k -clustering. By Lemma 5.3 and the existence of a pair of 3-bounded mapping and (4, 6)-bounded covering (see Lemma 5.10 and 5.13), the cost the returned k -clustering is not more than $84 \cdot \text{OPT}$ where OPT is the cost of an optimal α -fair k -clustering.

Finally, as we set $\varepsilon = O(1/k)$ and by Theorem 3.1 the runtime of the algorithm is $\tilde{O}(k^5 n^5)$. \square

6. Experiments

In this section, we provide an empirical evaluation of our local search based algorithm for α -fair k -clustering with k -median and k -means cost functions. We evaluate the empirical performance of the following approaches:

- **FAIRKCENTER (Jung et al., 2019).** First we consider the algorithm of (Jung et al., 2019), where they proposed to perform a binary search to find a value of $1 \leq \eta \leq 2$ for which the number of critical balls turns out to be exactly k . Then, the centers of these balls are the clustering centers.³ More precisely, for a given value of η , they run the algorithm of (Chan et al., 2006; Charikar et al., 2010), which is similar to Algorithm 1 but in line 6 a point x is marked as “covered” if $d(x, c) \leq \eta \cdot r_k(x)$ where c is the newly picked center.
- **Local Search with 1-Swap.** Second, we consider our local search algorithm as described in Algorithm 2. However to make it faster, we set t , the maximum size of swaps, equal to 1 instead of 4.
- **Greedy.** Finally as it seems a reasonable heuristic, we also consider the solution after the initialization step for our local search algorithm that is described in Algorithm 2, part I. This algorithm first finds critical balls and then includes their centers to S . Next, it goes through iterations until S becomes of size k . In each iteration it adds the point which furthest from the current set S . This algorithm is the initialization

³(Jung et al., 2019) remarked that while it is not always guaranteed that there exists such α that results in *exactly* k balls, this approach finds k centers on most natural datasets.

method we used for our local search algorithm and is described in Algorithm 2, part I.

Dataset. We consider three datasets from UCI Machine Learning Repository (2017)⁴ which are standard benchmarks for clustering algorithms and in particular they were used in the context of fair k -median clustering in (Chierichetti et al., 2017; Chen et al., 2019; Backurs et al., 2019; Bera et al., 2019; Huang et al., 2019). Formally, we consider the following datasets where in each of them we consider only numerical attributes:

- *Diabetes.* This dataset provides the information and outcome regarding patients related to diabetes from 1999 to 2008 at 130 hospitals across US⁵. Points in this datasets are in \mathbb{R}^2 and correspond to “age” and “time-in-hospital” attributes.
- *Bank.* This datasets corresponds to information from a Portuguese Bank⁶. Here, points live in \mathbb{R}^3 and corresponds to “age”, “balance” and “duration-of-account”.
- *Census.* The dataset is from 1994 US Census⁷ and here the selected attributes are “age”, “fnlwtg”, “education-num”, “capital-gain” and “hours-per-week”; points are in \mathbb{R}^5 .

Dataset	Dimension	# of Points	Aspect Ratio
Diabetes	2	101, 765	90.2
Bank	3	4, 520	13511.9
Census	5	32, 560	58685

Table 6.1. Some statistics about the datasets used in our experiments. Aspect ratio denotes the ratio between maximum distance and minimum distance.

Finally, in all our experiments we randomly sample a subset of size 1000 points from the data set and run our experiments on this sub-sample.

Experiment Setup. In our experiments, we follow the description of Algorithm 1 and 2. The only discrepancy is that instead of considering a point to be covered if it has a center within distance of 6 times its fair radius, in our implementation, we consider a point covered if it has a center within distance of 3 times its fair radius (see line 6 in Algorithm 1).

In all experiments, the input parameter α to our local search algorithms (i.e., the desired fairness guarantee) is the fair-

⁴<https://archive.ics.uci.edu/ml/datasets>

⁵<https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008>

⁶<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

⁷<https://archive.ics.uci.edu/ml/datasets/adult>

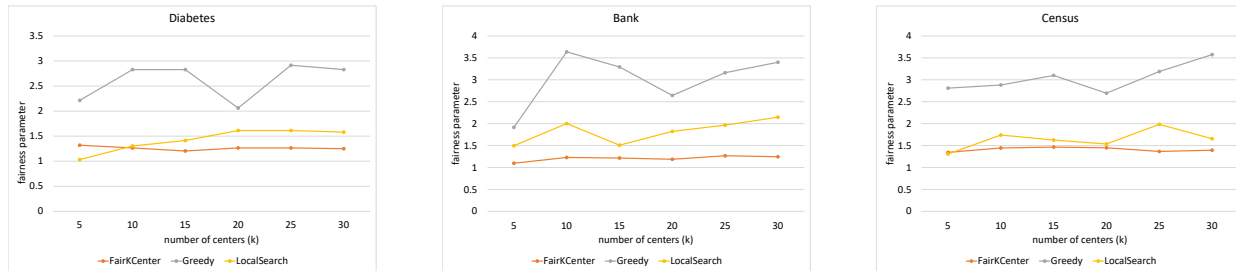


Figure 6.1. These plots compare the fairness guarantees of the solutions returned by each of the three algorithms we described for fair k -median in this section on the three datasets Diabetes, Bank and Census.

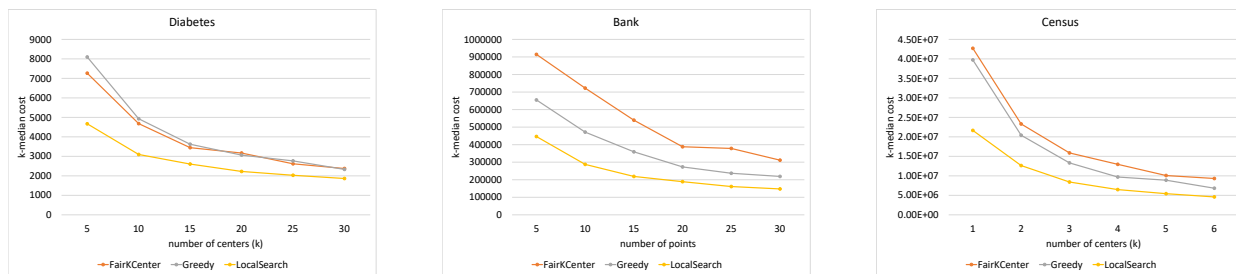


Figure 6.2. These plots compare the k -median costs of the solutions returned by each of the three algorithms we described for fair k -median in this section on the three datasets Diabetes, Bank and Census.

ness approximation $1 \leq \eta \leq 2$ returned by the FAIRKCENTER algorithm of (Jung et al., 2019).

Finally, we consider values of k to be in range 5 to 30 with steps of size 5 and draw our plots as a function of k .

Results. Figures 6.1 and 6.2 show empirical comparisons of the aforementioned algorithms, both in terms of fairness and the k -median cost of the solution. Our plots imply that local-search based algorithms perform reasonably well with respect to the notion of α -fairness: While its fairness guarantee is close to the fairness of FAIRKCENTER, it always exhibits a better performance in terms of k -median cost. More precisely, on average the local search algorithm reports a solution whose cost is better than the algorithm of (Jung et al., 2019) by a factor of 1.4, 2.25, and 1.93 while the reported solution has a worse fairness by a factor of 1.13, 1.5, and 1.16 for Diabetes, Bank and Census respectively.

We note that we observe a similar behavior of local search based algorithm for fair k -means. The plots are presented in Appendix C.

7. Conclusion and Future Direction

We presented the first bicriteria approximation algorithm for k -means, k -median and k -center (and more generally, any ℓ_p

norm cost function) with respect to the notion of individual fairness introduced by (Jung et al., 2019). In particular, Our analysis shows that the popular local search with constant size swaps achieves $(O(1), O(1))$ -approximation for fair k -median and fair k -means.

While our algorithm runs in polynomial time $\tilde{O}(k^5 n^5)$, an interesting direction is whether we can design a near-linear time algorithm for this problem presumably with worse approximation guarantees.

Acknowledgements

The authors would like to thank Konstantin Makarychev and Yury Makarychev for their helpful discussions on this work.

References

Abraham, S. S., P. D., and Sundaram, S. S. Fairness in clustering with multiple sensitive attributes. *arXiv preprint arXiv:1910.05113*, 2019.

Ahmadian, S., Epasto, A., Kumar, R., and Mahdian, M. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference*

- on *Knowledge Discovery & Data Mining*, pp. 267–275, 2019.
- Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. Local search heuristics for k -median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.
- Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. Scalable fair clustering. In *International Conference on Machine Learning*, pp. 405–413, 2019.
- Baharlouei, S., Nouiehed, M., and Razaviyayn, M. R\`enyi fair inference. *arXiv preprint arXiv:1906.12005*, 2019.
- Bera, S., Chakrabarty, D., Flores, N., and Negahbani, M. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*, pp. 4955–4966, 2019.
- Bercea, I. O., Groß, M., Khuller, S., Kumar, A., Rösner, C., Schmidt, D. R., and Schmidt, M. On the cost of essentially fair clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- Chan, T.-H. H., Dinitz, M., and Gupta, A. Spanners with slack. In *European Symposium on Algorithms*, pp. 196–207, 2006.
- Charikar, M., Makarychev, K., and Makarychev, Y. Local global tradeoffs in metric embeddings. *SIAM Journal on Computing*, 39(6):2487–2512, 2010.
- Chen, X., Fain, B., Lyu, L., and Munagala, K. Proportionally fair clustering. In *International Conference on Machine Learning*, pp. 1032–1041, 2019.
- Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pp. 5036–5044, 2017.
- Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.
- Chouldechova, A. and Roth, A. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.
- Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. Dimensionality reduction for k -means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 163–172, 2015.
- Davidson, I. and Ravi, S. Making existing clusterings fairer: Algorithms, complexity results and insights. Technical report, Technical report, University of California, Davis, CA., 2019.
- Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Dressel, J. and Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.
- Elzayn, H., Jabbari, S., Jung, C., Kearns, M., Neel, S., Roth, A., and Schutzman, Z. Fair algorithms for learning in allocation problems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 170–179, 2019.
- Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.
- Galindo, J. and Tamayo, P. Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. *Computational Economics*, 15(1-2):107–143, 2000.
- Gupta, A. and Tangwongsan, K. Simpler analyses of local search algorithms for facility location. *arXiv preprint arXiv:0809.2554*, 2008.
- Har-Peled, S. and Mahabadi, S. Near neighbor: Who is the fairest of them all? In *Advances in Neural Information Processing Systems*, pp. 13176–13187, 2019.
- Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pp. 3315–3323, 2016.
- Huang, L., Jiang, S., and Vishnoi, N. Coresets for clustering with fairness constraints. In *Advances in Neural Information Processing Systems*, pp. 7587–7598, 2019.
- Johnson, W. B. and Lindenstrauss, J. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
- Joseph, M., Kearns, M., Morgenstern, J. H., and Roth, A. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pp. 325–333, 2016.
- Jung, C., Kannan, S., and Lutz, N. A center in your neighborhood: Fairness in facility location. *arXiv preprint arXiv:1908.09041*, 2019.
- Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. A local search approximation algorithm for k -means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

- Kleinberg, J., Mullainathan, S., and Raghavan, M. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1): 237–293, 2018.
- Kleindessner, M., Awasthi, P., and Morgenstern, J. Fair k -center clustering for data summarization. In *International Conference on Machine Learning*, pp. 3448–3457, 2019a.
- Kleindessner, M., Samadi, S., Awasthi, P., and Morgenstern, J. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, pp. 3458–3467, 2019b.
- Makarychev, K., Makarychev, Y., and Razenshteyn, I. Performance of johnson-lindenstrauss transform for k -means and k -medians clustering. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1027–1038, 2019.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.
- Pessach, D. and Shmueli, E. Algorithmic fairness, 2020.
- Rösner, C. and Schmidt, M. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pp. 96:1–96:14, 2018.
- Schmidt, M., Schwiegelshohn, C., and Sohler, C. Fair coresets and streaming algorithms for fair k -means. In *International Workshop on Approximation and Online Algorithms*, pp. 232–251. Springer, 2019.